



**程序设计实验报告3**

学 院

专 业

年级班别

学 号

学生姓名

指导教师

2018年 10 月

# 实验3

## 实验目的

1. 能够定义和引用一维数组、二维数组和字符数组；
2. 能够利用数组来设计相关程序；
3. 掌握函数的定义和调用方法、变量的作用域和存储方式等概念；
4. 理解函数调用时参数传递的过程；
5. 能够区分值传递方式和地址传递；
6. 能够用函数实现模块化的程序设计。

## 实验工具

Tubro C 2.0/3.0

## 编写要求

对每个实验题目，需要提供如下内容：

1. 程序流程图（应采用NS图或传统流程图）；
2. 具体程序代码（具有良好的程序设计风格，如程序模块化、加注释、缩进的书写格式）；
3. 实验运行结果（对比测试用例）；
4. 对运行情况所做的分析以及调试程序所取得的经验收获。

## 排版要求

1. 标题四号宋体，正文小四宋体，段间距1.5倍；
2. 除封面以外，其他内容双面打印。

## 实验题目

1． 有两个整型数组，一个长度为6，一个长度为5,将两个数组合并,存放在a和b数组中，合并和存放规则如下

(1)先采用冒泡法将a数组按照升序进行排序，输出a数组的所有元素

(2)先采用冒泡法将b数组按照升序进行排序，输出b数组的所有元素

(3)将b数组的元素插入到a数组当中，若是最小放在最前面，否则插入到一个合适的位置

(4)若a数组不够存储，则存放到b数组中，剩下的数也是升序排序。

(5)不允许借组第三个数组来实现功能

(6)将合并后a数组和b数组分别输出,文件名取名为P1.c

比如：a {8,3,1,5,4,9} b {6,3,2,-1,10}

(1)排序 a {1,3,4,5,8,9} b{-1,2,3,6,10}

(2)插入合并a {-1,1,2,3,3,4} b{5,6,8,9,10}

测试用例：(1) a:{8,3,1,5,4,9} b:{6,3,2,-1,10}

合并后结果 a:{-1,1,2,3,3,4} b:{5,6,8,9,10}

(2) a:{11,8,7,6,4,3} b:{2,1,5,3,9}

合并后结果 a:{1,2,3,3,4,5} b:{6,7,8,9,11}

(3) a:{0,-3,8,100,20,30,}，b:{4,10,101,-5,0}

合并后结果 a:{-5,-3,0,0,4,8,}，b:{10,20,30,100,101}

**NS图**

**void sort(int \*a, int n);**

|  |  |  |  |
| --- | --- | --- | --- |
| int i, j, temp | | | |
| for（i= 0; i < n - 1; i++） | | | |
|  | for (j = 0; j < n – 1 - i; j++) | | |
|  | N  Y  a[j]>a[j+1] | |
| temp=a[i]  a[i]=a[j]  a[j]=temp |  |

**main函数**

|  |  |  |  |
| --- | --- | --- | --- |
| int i, j, k, temp, a[6], b[5] | | | |
| 输入数组a, b | | | |
| sort(a, 6） | | | |
| sort(b, 5) | | | |
| 输出数组a,b | | | |
| for (i = 0, i < 5; i++) | | | |
|  | for (j = 0; j < 6; j++) | | |
|  | N  Y  a[j]>b[i] | |
| temp=a[j]  a[j]=b[i]  b[i]=temp |  |
| sort(b, 5) | | | |
| 输出数组a,b | | | |

**代码：**

#include <stdio.h>

void sort(int \*, int);//需多次用到排序，另写一个函数会比较方便

int main()

{

int a[6], b[5], i, j, temp;

for (i = 0; i < 6; i++)

scanf("%d", &a[i]);

for (i = 0; i < 5; i++)

scanf("%d", &b[i]);

sort(a, 6);//对a,b进行冒泡排序

sort(b, 5);

for (i = 0; i < 6; i++)

printf("%d ", a[i]);

printf("\n");

for (i = 0; i < 5; i++)

printf("%d ", b[i]);

printf("\n");

for (i = 0; i < 5; i++)

{

for (j = 0; j < 6; j++)//遍历a，使得a中最大的元素交换到b中

{

if (a[j] > b[i])

{

temp = a[j];

a[j] = b[i];

b[i] = temp;

}

}

}

sort(b, 5);//重新对b进行冒泡排序

for (i = 0; i < 6; i++)

printf("%d ", a[i]);

printf("\n");

for (i = 0; i < 5; i++)

printf("%d ", b[i]);

printf("\n\n");

return 0;

}

void sort(int \*a, int n)

{

int i, j, temp;

for (i = 0; i < n - 1; i++)

{

for (j = 0; j < n - 1 - i; j++)

{

if (a[j] > a[j + 1])

{

temp = a[j];

a[j] = a[j + 1];

a[j + 1] = temp;

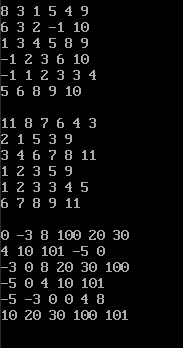
}

}

}

}

**运行结果：**



**总结：**

感觉这次实验也有些不足的地方，可以把合并操作也是要一个函数进行整合。